

Pan-private Algorithms: When Memory Does Not Help

Darakshan Mir* S. Muthukrishnan† Aleksandar Nikolov‡ Rebecca N. Wright§

September 9, 2010

Abstract

Consider updates arriving online in which the t th input is (i_t, d_t) , where i_t 's are thought of as IDs of users. Informally, a randomized function f is *differentially private* with respect to the IDs if the probability distribution induced by f is not much different from that induced by it on an input in which occurrences of an ID j are replaced with some other ID k . Recently, this notion was extended to *pan-privacy* where the computation of f retains differential privacy, even if the internal memory of the algorithm is exposed to the adversary (say by a malicious break-in or by fiat by the government). This is a strong notion of privacy, and surprisingly, for basic counting tasks such as distinct counts, heavy hitters and others, Dwork et al [4] present pan-private algorithms with reasonable accuracy. The pan-private algorithms are nontrivial, and rely on sampling.

We reexamine these basic counting tasks and show improved bounds. In particular, we estimate the distinct count $D^{(t)}$ to within $(1 \pm \epsilon)D^{(t)} \pm O(\text{polylog } m)$, where m is the number of elements in the universe. This uses suitably noisy statistics on sketches known in the streaming literature. We also present the first known lower bounds for pan-privacy with respect to a single intrusion. Our lower bounds show that, even if allowed to work with unbounded memory, pan-private algorithms for distinct counts can not be significantly more accurate than our algorithms. Our lower bound uses noisy decoding. For heavy hitter counts, we present a pan private streaming algorithm that is accurate to within $O(k)$ in worst case; previously known bound for this problem is arbitrarily worse. An interesting aspect of our pan-private algorithms is that, they deliberately use very small (polylogarithmic) space and tend to be streaming algorithms, even though using more space is not forbidden.

*mir@cs.rutgers.edu. Rutgers University. Work supported by NSF Awardd Number CCF-0728937, and U.S. DHS Award Number 2009-ST-061-CCI002,

†muthu@cs.rutgers.edu. Rutgers University. Work supported by NSF awards 0354690, 0414852 and 0916782, and DHS CCICADA.

‡anikolov@cs.rutgers.edu. Rutgers University

§rebecca.wright@rutgers.edu. Rutgers University

1 Introduction

Consider updates arriving online in which the t th input is (i_t, d_t) . Define input S_t as the first t updates, i.e. $(i_1, d_1), \dots, (i_t, d_t)$; i_t 's are IDs of users from Universe \mathcal{U} of size m . An example is to think of this as a “traffic log” where i_t is the ID of a user, $i_t \in \mathcal{U}$ and d_t is the time spent by the user at a particular website of interest; another example is to think of the input as a “payment log” where i_t is the ID of a merchant, $i_t \in \mathcal{U}$ and d_t the transaction value, which may be positive for sales and negative for refunds. A user may visit the site many times and a merchant may have many transactions; hence, same i_t 's may be seen several times. It is of great interest to maintain various statistics on such logs. For example,

- *distinct count*, $D^{(t)}$, is the number of distinct i_j 's seen before t th update;
- *heavy hitters count* $HH(k)$, informally, is the number of i 's that have large total d , $\sum_{j \leq t | i_j = i} d_j$ (a precise definition is presented later);
- *rarity ratio* $r(k) = \frac{|\{i | (\sum_{j \leq t | i_j = i} d_j) = k\}|}{D^{(t)}}$;
- *frequency moment* $F_k = \sum_{i \in \mathcal{U}} (\sum_{j \leq t | i_j = i} d_j)^k$;

and others. Normally, these statistics are trivial to maintain with an array \mathbf{a} of size m ($a_i = \sum_{j \leq t | i_j = i} d_j$), and some basic bookkeeping. These statistics – in one form or the other – have a long history, and are considered basic in data analysis tasks over the past few decades.

Our focus is on privacy, that is, how to maintain these statistics and still preserve the privacy of IDs involved. There are two concerns:

- *What if the output reveals something about the IDs?* For example, an adversary might estimate $D^{(t)}$ first, and then insert an $(i, 1)$ before determining D_{t+1} which will surely reveal if i was already in the input prior to t . Likewise one can devise insertion and query strategies that will reveal information about various IDs from other statistical queries.
- *What if the adversary gets access to the system and sees the internal memory used by the algorithm?* This might happen not only with intruders but may even be the outcome of a legal request which will force us to reveal all the stored information. In this case, with the trivial solution, a_i will end up revealing information about i . Of course, one could hash (encrypt) IDs and index in the hashed space. But when the memory is compromised, the hash (encryption) function will get revealed and will let the adversary decode by enumerating IDs. Often, sampling algorithms are used for providing statistical estimates, but these are vulnerable because when the internal memory is revealed, the sampled IDs are compromised.

To overcome the first concern, we can adopt the notion of *differential privacy* [6]. Let S'_t be the updates derived from S_t by replacing some occurrences of some ID j with occurrences of some other ID k . Informally, a randomized function f is *differentially private* with respect to the IDs if the probability distribution induced by $f(S_t)$ on the range of f is not much different from that induced by $f(S'_t)$ for any S'_t as defined above, and any t . For the first 3 statistics listed above, using known techniques, it is straightforward to get differentially private estimates; for frequency moments, one can look at a related function *cropped frequency moment* $T_k(\tau) = \sum_{i \in \mathcal{U}} \min\{(\sum_{j, i_j = i} d_j)^k, \tau\}$ that bounds what is known as the sensitivity of the function and get differentially private estimates.

The authors in [4] initiated the study that addresses the second concern above. In particular, they defined the notion of *pan-privacy*. Informally, S_t and S'_t should produce very similar distributions on *both* internal states as well as outputs. Without some “secret state” it might seem impossible to estimate statistics privately, but [4] showed that some of the statistics above can be estimated accurately. Their main results were for *streaming algorithms*, that use space polylogarithmic in m and other parameters. In particular, they showed pan-private streaming algorithms for rarity ratio, distinct count, cropped mean T_1 and a version of heavy hitters.

We are inspired by this work [4] and this emerging direction of pan-private algorithms [5] to revisit these problems. There are some outstanding fundamental questions:

- Is there a cost to pan-privacy, that is, are there problems for which pan-privacy provably needs more resources or loss of accuracy, compared to just differential privacy?
- What is the impact of memory in pan-privacy? Since the memory used by the algorithm may get revealed to an adversary, do pan-private algorithms use very small memory like in the streaming algorithms of [4], or can they use large memory to better encode information about the input and get better accuracy?
- Technically, [4] used samples and adapted techniques from *randomized response* [13] such as distorting counters with random shifts or using two distinct distributions. In contrast, in streaming [9], some of the most powerful algorithms use sketches that are linear projections of data along random directions. Do sketches provide improved or richer pan-private algorithms?

Our Contributions We address these questions and make the following main contributions. We focus on the basic model of pan-privacy as formulated by [4] where memory may be breached by an adversary once unannounced to the algorithm (and later comment on the variants of the model).

- *Distinct Counts.* We present a streaming algorithm that is ϵ -pan private and outputs an estimate $(1+\epsilon)D^{(t)} \pm \text{polylog}(m)$. It directly uses sketch known before based on stable distributions for estimating distinct counts [2], but maintains noisy versions. In fact, this approach is powerful and our pan-private algorithms even work for *turnstile* streams where d_i 's may be negative, the first pan-private algorithms to have this property. In contrast, best previous result for pan-private streaming estimation of distinct count outputs an estimate $D^{(t)} \pm \alpha m$ with constant probability, for only nonnegative updates [4]. Note that stable distribution based approach is known to yield streaming algorithms for F_k for $0 \leq k \leq 2$ [8], but this analogy does not work to get pan-private estimate of F_k (adapted as T_k); therefore, that it works for pan-private $k = 0$ (which is related to distinct counts) is very interesting.

We complement this result by showing lower bounds. Let \mathcal{A} be an online (not necessarily streaming) algorithm that on input S_t outputs $D^{(t)} \pm o(\sqrt{m})$ with small constant probability. Then \mathcal{A} is not ϵ -pan private for any constant ϵ . This is the first-known lower bound for any pan-private algorithm in this model. In fact, we develop an approach to showing lower bounds (which may be of independent interest in the future) that takes a copy of the memory by breaching the algorithm once, and then simulating the algorithm with random inputs in parallel with this seed memory like noisy decoding [3]. Our lower bound holds no matter the memory used by \mathcal{A} , even if the memory is $\Omega(m)$. Thus, \mathcal{A} need not be a streaming algorithm. Our lower bound is not like the ones in streaming literature where the lower bound is conditioned on using small space, or like in differentially private optimization [12] where one shows structural relationship between “near” configurations of inputs. We show this lower bound is essentially tight if \mathcal{A} is *not* streaming: we show a simple pan-private algorithm that outputs an estimate $D^{(t)} \pm o(\sqrt{m})$ with constant probability and maintains $O(m)$ memory. Further, we show a lower bound of $(1+\epsilon)D^{(t)} \pm \text{polylog}(m)$, essentially tight upto additive polylog terms with our streaming algorithm.

- *Heavy Hitters Count.* As is standard in streaming literature, we define $\text{HH}^{(t)}(k)$ as the number of IDs i with $\sum_{j: i_j=i} d_j \geq F_1^{(t)}/k$. In this notation, [4] approximates $\text{HH}(k)$ within an additive error of $O(\alpha m)$ for any constant α . However, m can far exceed k which is an upper bound on $\text{HH}(k)$.

We present a pan-private streaming algorithm that returns an estimate in $[(1-\epsilon)\text{HH}(k) - O(\sqrt{k}), \text{HH}(O(k^2)) + O(\sqrt{k})]$ (that is no worse than $O(k)$ approximation, upto additive errors), which is a significant improvement over [4]. We obtain this by first observing that with $O(m)$ space, we can provide an estimate $\text{HH}(k) \pm O(\sqrt{m})$, and then using this only on the space of all buckets in the Count-Min sketch [1] which uses much smaller space.

Some comments: (1) Both of our results above are obtained using sketches, which is different from use of samples thus far [4]. Also, we use full space versions on top of sketches to get best-known accuracies. (2) An interesting aspect of our pan-private algorithms is that, they deliberately use very small (polylogarithmic) space and are streaming even though using more space is not forbidden. (3) Our insights from above yield other upper bounds (pan-private streaming algorithms for T_k , inner products of vectors etc) and lower bounds

(inner products). (4) We are adapting pan-privacy model from [4] as a given and refer the readers to that original work for motivating and defending the model as well as discussion related to the model such as, what if a small amount of secret storage is allowed, or what if adversary is allowed to look at the memory multiple times or even continually and so on. For the purposes of this paper, the basic pan-privacy model is of great interest and there are fundamental technical problems that we address. (5) Likewise, the specific statistics we have considered have many applications that have been identified over the past decade from databases to data streams, compressed sensing and beyond [9]. We do not elaborate on this further, instead addressing how these problems can be solved. (6) Finally, we have focused on counts throughout. Many of these problems have a corresponding “list” version in which output comprises specific IDs. We have left it open to identify suitable pan-private versions of these problems.

Map. In Section 2, we introduce relevant definitions and notation. In Section 3, we present our upper and lower bounds for distinct count estimation. In Section 4, we present our upper bound for heavy hitter count estimation. In Section 5, we have concluding remarks with other extensions.

2 Preliminaries

2.1 Definitions and Notation

We are given a universe \mathcal{U} , where $|\mathcal{U}| = m$. An *update* is defined as an ordered pair $(i, d) \in \mathcal{U} \times \mathbb{Z}$. Consider a semi-infinite sequence of updates $(i_1, d_1), (i_2, d_2), \dots$; the *input* for all our algorithms consists of the first t updates, denoted $S_t = (i_1, d_1), \dots, (i_t, d_t)$. The *state* after t updates is an m -dimensional vector $\mathbf{a}^{(t)}$, indexed by the elements in \mathcal{U} (we will omit the superscript when it is clear from the context). The elements of the vector $\mathbf{a} = \mathbf{a}^{(t)}$, referred to as the *state vector*, are defined as follows:

$$a_i = \sum_{j: i_j = i} d_j.$$

We consider two models: the *cash register model* in which all updates are positive, i.e. $\forall j : d_j \geq 0$, and the *turnstile model* in which updates can be both positive (*inserts*), i.e. $d_j \geq 0$, and negative (*deletes*), i.e. $d_j < 0$. We note that the turnstile model has not been considered in pan privacy before.

Our algorithms *output* a real number which approximates one of the following statistics on S_t :

- *distinct count*: $D = D^{(t)} = |\{i \in \mathcal{U} : a_i \neq 0\}|$;
- *k-th frequency moment*: $F_k = F_k^{(t)} = \sum_{i \in \mathcal{U}} |a_i|^k$. This coincides with the L_k norm, $\|\mathbf{a}\|_k$ of the state vector \mathbf{a} and we will use either terms to facilitate exposition.
- *k-th cropped frequency moment*: $T_k(\tau) = T_k^{(t)}(\tau) = \sum_{i \in \mathcal{U}} \min\{|a_i|^k, \tau\}$.
- *cropped dot product*: Given two sequences of updates S_t and S'_t with state vectors \mathbf{a} and \mathbf{a}' , the cropped dot product is $(\mathbf{a} \cdot \mathbf{a}')(\tau) = \sum_{i \in \mathcal{U}} \min\{a_i a'_i, \tau\}$.
- *k-heavy hitters count*: $\text{HH}(k) = \text{HH}^{(t)}(k) = |\{i : |a_i| \geq F_1^{(t)}/k\}|$.

2.2 Differential Privacy

Dwork et al. [6] introduce the concept of differential privacy which operates on a data set consisting of rows of data, where each row consists of the data of an individual. Differential privacy provides a guarantee that the probability distribution on the outputs of a mechanism is “almost the same”, irrespective of whether an individual opts in to, or out of, the data set. Such a guarantee incentivizes participation of individuals in a database by assuring them of incurring very little risk by such a participation. Formally,

Definition 1 ([6]). A randomized function f provides ϵ -differential privacy if for all neighboring (differing in at most one row) data sets D and D' , and all $Y \subseteq \text{Range}(f)$,

$$\Pr[f(D) \in Y] \leq \exp(\epsilon) \times \Pr[f(D') \in Y].$$

One mechanism that [6] use to provide differential privacy is the so called “Laplacian noise method”, which depends on the *global sensitivity* of a function:

Definition 2 ([6]). *For $f : \mathcal{D} \rightarrow \mathbb{R}^d$, the global sensitivity of f is*

$$GS_f = \max_{D, D'} \|f(D) - f(D')\|_1$$

for all neighboring data sets D and D' .

The Laplace distribution with mean 0 and scale parameter b , denoted $Lap(b)$, has density function $p(x) = \frac{1}{2b} \exp(-|x|/b)$. The following theorem from [6] uses the Laplace distribution to construct a differentially private mechanism:

Theorem 1 ([6]). *For $f : \mathcal{D} \rightarrow \mathbb{R}$, mechanism \mathcal{M} that adds independently generated noise drawn from $Lap(GS_f/\epsilon)$ to the output preserves ϵ -differential privacy.*

2.3 Pan-privacy

While differential privacy provides meaningful guarantees to mitigate the risks of an individual being identified by participating in a data set, individuals might also be concerned about retaining similar guarantees even if the *internal state* is revealed, say, because of a subpoena. Mechanisms that achieve this property are called *pan-private* [4]. Pan privacy guarantees a participant that his/her risk of being identified by participating in a data set is very little even if there is an external intrusion on the data. Formally, consider two online updates $S = \{(i_1, d_1), \dots, (i_t, d_t)\}$ and $S' = \{(i'_1, d'_1), \dots, (i'_{t'}, d'_{t'})\}$ associated with state vectors \mathbf{a} and \mathbf{a}' respectively.

Definition 3. *S and S' are said to be neighbors if there exists a (multi)set of updates in S indexed by $K \subseteq [t]$ that update the same ID $i \in \mathcal{U}$, and there exists a (multi)set of updates in S' indexed by $K' \subseteq [t']$ that updates some $j (\neq i) \in \mathcal{U}$ such that $\sum_{k \in K} d_k = \sum_{k \in K'} d'_k$ and for all other updates in S and S' indexed by $Q = [t] - K$ and $Q' = [t'] - K'$ respectively,*

$$\forall i \in \mathcal{U} \quad \sum_{k \in Q, s.t. \ i_k = i} d_k = \sum_{k \in Q', s.t. \ i'_k = i} d'_k$$

Notice that in the definition above t and t' don't have to be equal because we allow the d_i 's to be integers. The definition ensures that two inputs are neighbors if some of the occurrences of an ID in S is replaced by some other ID in S' and everything else essentially stays the same except (a) the order may be arbitrarily different and (b) the updates can be broken up since they are not constrained to be 1's. The neighbor relation preserves the first frequency moment of the sequence of updates, considered to be public information. Also, the graph induced by the neighbor relation on any set of sequences with the same first frequency moment is connected.

Definition 4 (User level pan-privacy[4]). *Let \mathbf{Alg} be an algorithm. Let I denote the set of internal states of the algorithm, and let σ the set of possible output sequences. Then algorithm \mathbf{Alg} mapping input prefixes to the range $I \times \sigma$, is pan-private (against a single intrusion)¹ if for all sets $I' \subseteq I$ and $\sigma' \subseteq \sigma$, and for all pairs of user-level neighboring data stream prefixes S and S'*

$$\Pr[\mathbf{Alg}(S) \in (I', \sigma')] \leq e^\epsilon \Pr[\mathbf{Alg}(S') \in (I', \sigma')]$$

where the probability spaces are over the coin flips of the algorithm \mathbf{Alg} .

¹See [4] for discussion about multiple intrusions.

3 Distinct Count Estimation

In this section we present upper and lower bounds for the problem of pan-private estimation of the distinct count statistic $D^{(t)}$. We utilize a sketching approach based on a stable distribution. In contrast with the sampling approach of Dwork et al. [4], the sketching approach works in the more general turnstile model and for the usual range of $D^{(t)}$ achieves significantly better accuracy. We present our algorithm for distinct count estimation as evidence of the usefulness of the sketching approach for designing pan-private algorithms.

We compliment our upper bound with lower bounds based on noisy decoding. Our results present the first lower bounds against pan-private algorithms that allow a single intrusion.

3.1 Upper Bounds

Consider the turnstile model where the d_j 's could either be positive or negative, and assume an upper bound on the absolute value of each element of the state vector: $\forall i \in \mathcal{U}, |a_i| < Z$. We are interested in a pan-private computation of $D^{(t)} = \{i | \mathbf{a}^{(t)}[i] \neq 0\}$. Note that where the superscripts don't appear a time slice of t is implicit. Recall that the L_p norm of a vector \mathbf{a} is $\|\mathbf{a}\|_p = (\sum_i |a_i|^p)^{\frac{1}{p}}$.

3.2 Prior Approach in Streaming Algorithms

[2] show that, for sufficiently small p ($0 < p < \epsilon / \log Z$)

$$D^{(t)} \leq \sum_i |a_i|^p \leq (1 + \epsilon) D^{(t)}. \quad (1)$$

Hence, it suffices to estimate the L_p norm of \mathbf{a} for certain small p for estimating the distinct counts. For this purpose they use what are called *stable distributions*.

Stable distributions and their use in sketches A distribution \mathcal{P} over \mathbb{R} is said to be p -stable, if there exists $p \geq 0$ such that for any n real numbers b_1, \dots, b_m and i.i.d. variables Y_1, \dots, Y_m with distribution \mathcal{P} , the random variable $\sum_i b_i Y_i$ has the same distribution as the random variable $(\sum_i |b_i|^p)^{1/p} Y$, where Y is a random variable with distribution \mathcal{P} [11]. Let X be a matrix of random values of dimension $m \times r$, where each entry of the matrix $X_{i,j}$, $1 \leq i \leq m$, and $1 \leq j \leq r$, is drawn independently from a random stable distribution with parameter p , with p as small as possible. The *sketch vector* $\text{sk}(\mathbf{a})$ is defined as the dot product of matrix X^T with \mathbf{a} , so

$$\text{sk}(\mathbf{a})_j = \sum_{i=1}^m X_{i,j} a_i = X_j \cdot \mathbf{a},$$

where X_j is a m -dimensional vector composed of the following elements: $(X_{1,j}, X_{2,j}, \dots, X_{m,j})$.

From the property of stable distributions we know that each entry of $\text{sk}(\mathbf{a})$ is distributed as $(\sum_i |a_i|^p)^{1/p} X_0$, where X_0 is a random variable chosen from a p -stable distribution. The sketch is used to compute $\sum_i |a_i|^p$ for $0 < p < \epsilon / \log Z$, from which we can approximate $D^{(t)}$ up to a $(1 + \epsilon)$ factor. By construction, any $\text{sk}(\mathbf{a})_j$ can be used to estimate L_p^p . [2] obtain a good estimator for $(\sum_i |a_i|^p)$ by taking the median of all entries $|\text{sk}(\mathbf{a})_j|^p$ over j :

Lemma 1 ([2]). *With probability $1 - \delta$ if $r = O(1/\epsilon^2 \cdot \log(1/\delta))$,*

$$(1 - \epsilon)^p \text{median}_j |\text{sk}(\mathbf{a})_j|^p \leq \text{median}_j |X_0|^p \left(\sum_i |a_i|^p \right) \leq (1 + \epsilon)^p \text{median}_j |\text{sk}(\mathbf{a})_j|^p$$

where $\text{median}_j |X_0|^p$, is the median of absolute values (raised to the power p) from a p -stable distribution.

Using the results of Equation 1 and Lemma 1 Cormode et al. [2] prove that:

Theorem 2 ([2]). *The computation of a sketch $\text{sk}(\mathbf{a})$ of online data described by a state vector \mathbf{a} that requires space $O(1/\epsilon^2 \cdot \log(1/\delta))$ allows an approximation of $D^{(t)}$ within a factor of $1 \pm \epsilon$ of the true answer with probability $1 - \delta$.*

Maintaining the sketch under updates As updates arrive, the sketch vector is built progressively. It is initialized to be the zero vector, and on receiving tuple (i, d_k) , the update is done by adding d_k times $X_{i,j}$ to each entry $\text{sk}(\mathbf{a})_j$, $\forall j \in [r]$ of the sketch vector. That is,

$$\forall j \in [r] : \text{sk}(\mathbf{a})_j \leftarrow \text{sk}(\mathbf{a})_j + d_k X_{i,j}.$$

In order to avoid percomputing and storing all the values $X_{i,j}$ Cormode et al. [2] generate the random variables $X_{i,j}$ from a stable distribution on the fly by using i to seed a pseudo-random number generator $\text{random}()$. These pseudo-randomly generated numbers are then used to generate a sequence of p -stable distributed random variables using a (deterministic) function $\text{stable}(r_1, r_2, p)$, where r_1 and r_2 are pseudo-random variables in the range $[0 \dots 1]$ drawn from $\text{random}()$. The function is defined as follows: first define a quantity $\theta = \pi(r_1 - 1/2)$. Now,

$$\text{stable}(1/2 + \theta, r_2, p) = \frac{\sin p\theta}{\cos^{1/p} \theta} \left(\frac{\cos(\theta(1-p))}{-\ln r_2} \right)^{\frac{1-p}{p}}.$$

Since each time the same seed i is used, this ensures that $X_{i,j} = \text{stable}(r_1, r_2, p)$ takes the same value each time it is used. We will find this technique useful for our own purpose of precomputing the global sensitivity of a sketch in the next section.

3.3 Pan-Private Algorithm

To get pan-privacy, we maintain these (approximate) sketches in a differentially-private way. In particular, we maintain a noisy sketch vector where each element of the sketch vector has noise added according to the sensitivity method of [6].

Adding Laplacian noise to the sketches. The global sensitivity of a sketch $\text{sk}(\mathbf{a})_j$, (GS_j) from Definition 2 is

$$GS_j = 2 \cdot Z \|X_j\|_\infty.$$

Consider state vectors \mathbf{a} and \mathbf{a}' corresponding to two neighboring sequences of online updates S and S' respectively. From Definition 3 there exists some $i \in [n]$ and some $k \neq i \in [n]$, such that some occurrences of i in the sequence of updates in S is replaced by some occurrences of k to get S' . This means that $a_i \neq a'_i$ and $a_k \neq a'_k$, and for any other l not equal to i or k , $a_l = a'_l$. So, for any neighboring S and S' ,

$$\|X_j \cdot \mathbf{a} - X_j \cdot \mathbf{a}'\|_1 \leq |X_{i,j}a_i - X_{i,j}a'_i + X_{k,j}a_k - X_{k,j}a'_k| \leq 2 \cdot Z \|X_j\|_\infty.$$

From [6], it will follow that we need to add Laplacian noise based on this sensitivity to have a differentially private description of the state at any point, which is pan-private with respect to a single intrusion. Since the elements of X_j are random quantities independent of the data, we can compute the L_∞ norm of the actual vector that we end up using without compromising on privacy. However, the challenge is that $\|X_j\|_\infty$ is not known in advance. The use of index i to seed the pseudorandom generator and use of the pseudorandomly generated values to generate the $X_{i,j}$'s, means that this challenge can be solved by computing $\|X_j\|_\infty$, $\forall j$ before the onset of our algorithm (shown in Algorithm 1). Also to use the result of Lemma 1, the value of $\text{median}|X_0|^p$, the median of absolute values from a p -stable distribution, needs to be computed. This is also done numerically in advance in [2], and then the final result is scaled by this constant factor denoted as $\text{sf}(p)$.

Algorithm 1 modifies the algorithm in [2] by maintaining α -differentially private sketches of the stream vector \mathbf{a} .

Each sketch is initialized with a noisy value drawn from the appropriate Laplace distribution. Formally, let $\text{sk}(\mathbf{a})^{priv}_j = \text{sk}(\mathbf{a})_j + \eta_j$ where η_j is a random variable drawn from a Laplacian distribution with mean 0 and scaling factor of GS_j/α . Here α is the privacy parameter. Since we maintain r sketches of the data, Algorithm 1 gives us an overall privacy of $\alpha' = \alpha r$ as per the composition theorem [6]:

Theorem 3 ([6]). *Given mechanisms \mathcal{M}_i , $i \in [r]$ each of which provide α_i -differential privacy, then the overall mechanism \mathcal{M} that consists of a composition of these r mechanisms, provides $\left(\sum_{i \in [r]} \alpha_i\right)$ -differential privacy.*

Algorithm 1 Pan-private approximation of $D^{(t)}$

INPUT: privacy parameter α , $0 < p < \epsilon/Z < 1$, $\|X_j\|_\infty \forall j \in [r]$ computed off-line, an r -dimensional noise vector $\boldsymbol{\eta}$, where $\eta_j \sim \text{Lap}(\frac{2\|X_j\|_\infty Z}{\alpha})$, $\text{sf}(p) = \text{median } |X_0|^p$ also computed off-line numerically.

initialize the r -dimensional sketch vector $\text{sk}(\mathbf{a})^{\text{priv}}$, such that $\text{sk}(\mathbf{a})_j^{\text{priv}} = \eta_j$

for all tuples (i, d_t) **do**

 initialize *random* with i

for all $j = 1$ to r **do**

$r1 = \text{random}()$

$r2 = \text{random}()$

$\text{sk}(\mathbf{a})_j^{\text{priv}} = \text{sk}(\mathbf{a})_j^{\text{priv}} + d_t * \text{stable}(r1, r2, p)$

end for

end for

return $\tilde{D} = \text{median}_j \left(\left| \text{sk}(\mathbf{a})_j^{\text{priv}} \right|^p \right) * \text{sf}(p)$

Our main result for distinct count estimation is to prove that \tilde{D} returned by Algorithm 1 provides an α' -differentially private approximation of $D^{(t)}$:

Theorem 4. *With probability $1 - (r + 1)\delta$, Algorithm 1 computes an α' -pan-private approximation \tilde{D} of $D^{(t)}$ such that*

$$(1 - \epsilon)D^{(t)} - O\left(\text{poly}\left(\log(m) \cdot (1 + \epsilon) \log\left(\frac{1}{\delta}\right) \frac{1}{\alpha'}\right)\right) \leq \tilde{D} \leq (1 + \epsilon)D^{(t)} + O\left(\text{poly}\left(\log(m) \cdot (1 + \epsilon) \log\left(\frac{1}{\delta}\right) \frac{1}{\alpha'}\right)\right)$$

We will need Claim 1, and Lemmas 3 and 2 for this purpose:

Claim 1. *For any two real numbers x and y and for any $p \in [0, 1]$, we have*

$$|x|^p - |y|^p \leq |x + y|^p \leq |x|^p + |y|^p$$

Proof. First, assume x and y are either both positive or negative. For any $x, y \in \mathbb{R}^+$ consider functions $g_{x,y}(p) = x^p + y^p$ and $f_{x,y}(p) = (x + y)^p$. At $p = 1$, $\forall x, y \in \mathbb{R}^+$, the two functions intersect as $x^1 + y^1 = (x + y)^1$. At $p = 0$, $g_{x,y}(p) > f_{x,y}(p)$. We want to prove that for $p \in [0, 1]$, $g_{x,y}(p) \geq f_{x,y}(p)$. For convenience, we drop the subscript x, y .

WLOG assume $x > y$, then $f(p) = x^p \left(1 + \frac{y}{x}\right)^p$ and $g(p) = x^p \left(1 + \left(\frac{y}{x}\right)^p\right)$ So,

$$\frac{f(p)}{g(p)} = \frac{\left(1 + \frac{y}{x}\right)^p}{1 + \left(\frac{y}{x}\right)^p}$$

The numerator

$$\left(1 + \frac{y}{x}\right)^p < 1 + \frac{y}{x}, \text{ for } p \in [0, 1], \forall \frac{y}{x} < 1$$

The denominator

$$1 + \left(\frac{y}{x}\right)^p > 1 + \frac{y}{x}, \text{ for } p \in [0, 1], \forall \frac{y}{x} < 1$$

$$\implies \frac{f(p)}{g(p)} < \frac{1 + \frac{y}{x}}{1 + \frac{y}{x}} = 1$$

So for any $x, y \in \mathbb{R}^+$, we have $f_{x,y}(p) < g_{x,y}(p)$, for $p \in [0, 1]$. Similarly, assume x is positive and y is negative, and WLOG assume $a = |x| > |y| = b$. Then $|x + y| = a - b$, and we can similarly prove that for $a, b \in \mathbb{R}^+$,

$$(a - b)^p \geq a^p - b^p$$

□

Lemma 2. With probability $1 - \delta$, for any $j \in [r]$, with $0 < p < \epsilon/Z < 1$

$$\left| \text{sk}(\mathbf{a})_j \right|^p - \xi \leq \left| \text{sk}(\mathbf{a})_j^{\text{priv}} \right|^p \leq \left| \text{sk}(\mathbf{a})_j \right|^p + \xi$$

where $\xi = \left(\left(\frac{2 \cdot Z \max_j \|X_j\|_\infty}{\alpha} \log\left(\frac{1}{\delta}\right) \right)^p \right)$

Proof. We have $\left| \text{sk}(\mathbf{a})_j^{\text{priv}} \right|^p = \left| \text{sk}(\mathbf{a})_j + \eta_j \right|^p$. From Claim 1, we have:

$$\left| \text{sk}(\mathbf{a})_j \right|^p - |\eta_j|^p \leq \left| \text{sk}(\mathbf{a})_j^{\text{priv}} \right|^p \leq \left| \text{sk}(\mathbf{a})_j \right|^p + |\eta_j|^p.$$

Also since η_j is drawn from a Laplacian distribution, we know that with probability $1 - \delta$, $|\eta_j| < \frac{GS_j}{\alpha} \cdot \log\left(\frac{1}{\delta}\right) \leq \frac{2Z \max_j \|X_j\|_\infty}{\alpha} \cdot \log\left(\frac{1}{\delta}\right)$. \square

Since Algorithm 1 computes \tilde{D} by taking the (scaled) median of the $\text{sk}(\mathbf{a})_j^{\text{priv}}$'s and Lemma 1 relates the median of the $\text{sk}(\mathbf{a})_j$'s to the $\sum_i |a_i|^p$, we need to bound $\text{median}_j \text{sk}(\mathbf{a})_j^{\text{priv}}$ in terms of $\text{median}_j \text{sk}(\mathbf{a})_j$.

Lemma 3. Let x_1, \dots, x_r and y_1, \dots, y_r be two sequences of real numbers satisfying $\forall i : x_i - E \leq y_i \leq x_i + E$. Then

$$\text{median}_i x_i - E \leq \text{median}_i y_i \leq \text{median}_i x_i + E.$$

Proof. Assume, WLOG, that x_1, \dots, x_r are sorted in increasing order and $\text{median}_i x_i = x_{\lceil r/2 \rceil}$. Let $\text{median}_i y_i = y_j$. We will prove that $y_j \geq x_{\lceil r/2 \rceil} - E$, and the other side of the inequality will follow by an analogous argument.

If $j \geq \lceil r/2 \rceil$, then $y_j \geq x_j - E \geq x_{\lceil r/2 \rceil} - E$. Therefore, we may assume $j < \lceil r/2 \rceil$. Because y_j has rank $\lceil r/2 \rceil$ in y_1, \dots, y_r , there exist indices $k_1, \dots, k_d > j$, where $d = \lceil r/2 \rceil - j$, s.t. $y_{k_1}, \dots, y_{k_d} \leq y_j$. At least one of k_1, \dots, k_d is greater than or equal to $\lceil r/2 \rceil$; let the smallest such index be ℓ . Then we have,

$$y_j \geq y_\ell \geq x_\ell - E \geq x_{\lceil r/2 \rceil} - E.$$

\square

Now we prove that \tilde{D} , returned by Algorithm 1 gives a good approximation to $D^{(t)}$:

Lemma 4. Algorithm 1 computes an α' -pan private approximation of $D^{(t)}$ using space $O(1/\epsilon^2 \log(1/\delta))$. The approximation guarantee is: With probability at least $1 - (r + 1)\delta$ and with ξ as in Lemma 2,

$$(1 - \epsilon)D^{(t)} - \xi \cdot \text{sf}(p) \leq \tilde{D} \leq (1 + \epsilon)D^{(t)} + \xi \cdot \text{sf}(p)$$

where $r = O(1/\epsilon^2 \cdot \log 1/\delta)$.

Proof. Since each sketch $\text{sk}(\mathbf{a})_j^{\text{priv}}$ is α -differentially private according to the sensitivity method of [6], and we have r such sketches, the over all privacy of the Algorithm is $\alpha r = \alpha'$. Each sketch is a differentially private description of the state and hence the algorithm achieves α' pan-privacy. Now we prove the approximation guarantee:

We have $\tilde{D} = \text{median}_j \text{sk}(\mathbf{a})_j^{\text{priv}} \cdot \text{sf}(p)$. Using Lemma 2 we have with probability at least $1 - r \cdot \delta \forall j$ simulatenously:

$$\left| \text{sk}(\mathbf{a})_j \right|^p - \xi \leq \left| \text{sk}(\mathbf{a})_j^{\text{priv}} \right|^p \leq \left| \text{sk}(\mathbf{a})_j \right|^p + \xi.$$

From Lemma 3, we have with probability at least $1 - r\delta$:

So we have with probability $1 - r\delta$

$$\text{median}_j \left| \text{sk}(\mathbf{a})_j \right|^p - \xi \leq \text{median}_j \left| \text{sk}(\mathbf{a})_j^{\text{priv}} \right|^p \leq \text{median}_j \left| \text{sk}(\mathbf{a})_j \right|^p + \xi.$$

Using Lemma 1 and Equation 1 and noting that $\alpha' = \alpha r$, the result follows. \square

Since $p < \epsilon / \log Z < 1$, and r , the number of sketches is polylogarithmic in m , and $\text{sf}(p)$ is a constant, from Lemma 4, we have:

Theorem 5. *With probability $1 - (r + 1)\delta$, Algorithm 1 computes an α' -pan-private approximation \tilde{D} of $D^{(t)}$ such that*

$$(1 - \epsilon)D^{(t)} - O\left(\text{poly}\left(\log(m) \cdot (1 + \epsilon) \log\left(\frac{1}{\delta}\right) \frac{1}{\alpha'}\right)\right) \leq \tilde{D} \leq (1 + \epsilon)D^{(t)} + O\left(\text{poly}\left(\log(m) \cdot (1 + \epsilon) \log\left(\frac{1}{\delta}\right) \frac{1}{\alpha'}\right)\right)$$

Proof. We have,

$$\xi \cdot \text{sf}(p) = O\left(\left(\frac{1}{\alpha'} r \cdot 2Z \max_j \|X_j\|_\infty \log\left(\frac{1}{\delta}\right)\right)^p\right)$$

and $Z^p < e^\epsilon$, which for small ϵ is less than $(1 + \epsilon)$. Since r is polylog in m and $\max_j \|X_j\|_\infty$ is a constant, the result follows. \square

In fact, this algorithm is a streaming algorithm since it stores polylogarithmic in m space and takes time polylogarithmic in m per new update. Technically, it works in the *turnstile* model since d_j may be positive or negative, the first such pan-private streaming algorithm [9].

The best previous result for pan-private distinct count estimation is due to Dwork et al. [4]. Their algorithm outputs and estimate in $[D^{(t)} - \alpha'm, D^{(t)} + \alpha'm]$ with probability $1 - \delta$ for any constant α and δ . By extending their techniques and running their algorithm in full space, we can get an estimate in $[D^{(t)} - O(\sqrt{m}), D^{(t)} + O(\sqrt{m})]$ with constant probability (see Section 4). Our sketching algorithm achieves a significantly smaller error whenever $D^{(t)} = o(\sqrt{m})$; we note that in practice the distinct counts statistic is usually much smaller than the size of the universe.

3.4 Lower bounds

Next we present lower bounds against pan-private algorithms that allow a single intrusion. These are the first such lower bounds in the literature and may be of independent interest.

We show that if only an additive approximation is allowed, the full space extension of Dwork et al.'s algorithm for distinct count estimation, as presented in Section 4, is optimal. Thus, the multiplicative approximation factor in the analysis of our sketching distinct counts algorithm is necessary. Furthermore, by proving a new noisy decoding theorem, we show that our sketching algorithm gives an almost optimal bi-approximation guarantee. Interestingly, our lower bounds make no assumptions on the space complexity of the algorithm, and yet the (almost) optimal algorithm happens to use polylogarithmic space.

Dinur-Nissim Style Decoding Our lower bounds utilize a decoding algorithm of the style introduced in a privacy context by Dinur and Nissim [3]. Informally, we argue that the (private) state of an accurate pan private algorithm can be used to recover the majority of the algorithm's input. First, we introduce the decoding results we will use.

Theorem 6 ([3]). *Let $\mathbf{x} \in \{0, 1\}^n$. For any ϵ and $n \geq n_\epsilon$, the following holds. Given $O(n \log^2 n)$ random strings $\mathbf{q}_1, \dots, \mathbf{q}_t \in_R \{0, 1\}^n$, and approximate answers $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_t$ s.t. $\forall i \in [t] : |\mathbf{x} \cdot \mathbf{q}_i - \tilde{\mathbf{a}}_i| = o(\sqrt{n})$, there exists an algorithm that outputs a string $\tilde{\mathbf{x}} \in \{0, 1\}^n$ and except with negligible probability $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \leq \epsilon n$.*

In follow up work, [7] strengthened the above and showed that decoding is possible even when a constant fraction of the queries are inaccurate.

Theorem 7 ([7]). *Given $\rho < \rho^*$, where ρ^* is a constant approximately equal to 0.239, there exists a constant ϵ s.t. the following holds. Let $\mathbf{x} \in \{0, 1\}^n$. There exists a matrix $A \in \{-1, 1\}^{n \times m}$ for some $m = O(n)$ and an efficient algorithm \mathcal{A} , s.t. on input $b \in \mathbb{N}^m$, satisfying $|\{i : |(A\mathbf{x} - \tilde{b})_i| > \alpha\}| \leq \rho$, \mathcal{A} outputs $\tilde{\mathbf{x}} \in \{0, 1\}^n$ and with probability $1 - e^{-O(m)}$, $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \leq \epsilon \alpha^2$*

Next we will prove a result that is similar to Dinur and Nissim's but uses "union queries" as opposed to dot product queries.

Theorem 8. Let $\mathbf{x} \in \{0,1\}^n$, $\|\mathbf{x}\|_0 \leq C \log^c n$ for some constants c and C . For any ϵ and $n \geq n_\epsilon$ the following statement holds. There exists $n^{O(\log^c n)}$ binary strings $\mathbf{q}_1, \dots, \mathbf{q}_t \in \{0,1\}^n$ and an algorithm \mathcal{A} such that given answers $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_t$ satisfying

$$\forall i : (1 - \alpha_1) \|\mathbf{x} + \mathbf{q}_i\|_0 - \alpha_2 \leq \tilde{\mathbf{a}}_i \leq (1 + \alpha_1) \|\mathbf{x} + \mathbf{q}_i\|_0 + \alpha_2$$

for $\alpha_2 = o(\log^c n)$, \mathcal{A} outputs $\tilde{\mathbf{x}}$ with $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \leq \frac{16(\alpha_1 + \epsilon)}{1 - \alpha_1} C \log^c n$.

Proof. Let L be an upper bound on $\|\mathbf{x}\|_0$, i.e. $L = C \log^c n$. The set of queries is $\mathbf{q}_0 = (0, \dots, 0)$, and $\mathbf{q}_1, \dots, \mathbf{q}_t$ are the indicator vectors of all subsets of $[n]$ of size at most L . The algorithm outputs any string $\tilde{\mathbf{x}}$ s.t. $\|\tilde{\mathbf{x}}\|_0 \leq L$ and $\tilde{\mathbf{x}}$ satisfies all the following constraints:

$$\forall i : (1 - \alpha_1) \|\tilde{\mathbf{x}} + \mathbf{q}_i\|_0 - \alpha_2 \leq \tilde{\mathbf{a}}_i \leq (1 + \alpha_1) \|\tilde{\mathbf{x}} + \mathbf{q}_i\|_0 + \alpha_2$$

Clearly the algorithm terminates, as at least one string, i.e. \mathbf{x} satisfies all constraints. Choose ϵ so that $\alpha_2 \leq \epsilon L$. Next we argue that if $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 > \frac{16(\alpha_1 + \epsilon)}{1 - \alpha_1} L$, at least one of the above constraints is violated.

We will consider several cases. Let $b = \frac{2(\alpha_1 + \epsilon)}{1 - \alpha_1} L$. Assume first that $\|\mathbf{x}\|_0 - \|\tilde{\mathbf{x}}\|_0 > b$. Then,

$$\begin{aligned} \tilde{\mathbf{a}}_0 &\geq (1 - \alpha_1) \|\mathbf{x}\|_0 - \alpha_2 \\ &> (1 - \alpha_1) (\|\tilde{\mathbf{x}}\|_0 + b) - \alpha_2 \\ &\geq (1 + \alpha_1) \|\tilde{\mathbf{x}}\|_0 + \alpha_2 - (2\alpha_1 \|\tilde{\mathbf{x}}\|_0 + 2\alpha_2 - (1 - \alpha_1)b) \\ &\geq (1 + \alpha_1) \|\tilde{\mathbf{x}}\|_0 + \alpha_2 - (2(\alpha_1 + \epsilon)L - (1 - \alpha_1)b) \\ &\geq (1 + \alpha_1) \|\tilde{\mathbf{x}}\|_0 + \alpha_2. \end{aligned}$$

We have shown that a constraint is violated by $\tilde{\mathbf{x}}$ in this case. The case $\|\tilde{\mathbf{x}}\|_0 - \|\mathbf{x}\|_0 > b$ is argued analogously.

Finally, assume that $\|\tilde{\mathbf{x}}\|_0 - \|\mathbf{x}\|_0 \in [-b, b]$. Let \mathbf{q}' be the indicator vector of the set $\{i : \mathbf{x}_i = 0, \tilde{\mathbf{x}}_i = 1\}$, and, similarly, let \mathbf{q}'' be the indicator vector of the set $\{i : \mathbf{x}_i = 1, \tilde{\mathbf{x}}_i = 0\}$. Since, by assumption $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 = \|\mathbf{q}'\|_0 + \|\mathbf{q}''\|_0 > \frac{16(\alpha_1 + \epsilon)}{1 - \alpha_1} L$, it follows that $\max(\|\mathbf{q}'\|_0, \|\mathbf{q}''\|_0) > \frac{8(\alpha_1 + \epsilon)}{1 - \alpha_1} L$. Assume, without loss of generality, that $\|\mathbf{q}'\|_0 > 8 \frac{8(\alpha_1 + \epsilon)}{1 - \alpha_1}$. We have the following identities:

$$\begin{aligned} \|\mathbf{q}' + \mathbf{x}\|_0 &= \|\mathbf{x}\|_0 + \|\mathbf{q}'\|_0 \\ \|\mathbf{q}' + \tilde{\mathbf{x}}\|_0 &\leq \|\tilde{\mathbf{x}}\|_0 + \|\mathbf{q}'\|_0 - \frac{4(\alpha_1 + \epsilon)}{1 - \alpha_1} L \\ \|\mathbf{q}' + \mathbf{x}\|_0 - \|\mathbf{q}' + \tilde{\mathbf{x}}\|_0 &\geq \frac{4(\alpha_1 + \epsilon)}{1 - \alpha_1} L - b \end{aligned}$$

Let $\tilde{\mathbf{a}}$ be the approximate answer to the query $\|\mathbf{q}' + \mathbf{x}\|_0$.

$$\begin{aligned} \tilde{\mathbf{a}} &\geq (1 - \alpha_1) \|\mathbf{x} + \mathbf{q}'\|_0 - \alpha_2 \\ &> (1 - \alpha_1) (\|\tilde{\mathbf{x}} + \mathbf{q}'\|_0 + 2b - \frac{8(\alpha_1 + \epsilon)}{1 - \alpha_1} L) - \alpha_2 \\ &= (1 + \alpha_1) \|\tilde{\mathbf{x}} + \mathbf{q}'\|_0 + \alpha_2 - (2\alpha_1 \|\tilde{\mathbf{x}} + \mathbf{q}'\|_0 + 2\alpha_2 + 2(1 - \alpha_1)b - 8(\alpha_1 + \epsilon)L) \\ &\geq (1 + \alpha_1) \|\tilde{\mathbf{x}} + \mathbf{q}'\|_0 - \alpha_2 - (2\alpha_1 L + 2\epsilon L + 4(\alpha_1 + \epsilon)L - 8(\alpha_1 + \epsilon)L) \\ &\geq (1 + \alpha_1) \|\tilde{\mathbf{x}} + \mathbf{q}'\|_0 - \alpha_2 \end{aligned}$$

Therefore, the constraint is violated and this completes the proof. \square

Lower Bounds from Noisy Decoding We introduce our approach to proving lower bounds for pan-private algorithms using the most direct argument first: a lower bound against dot product. We introduce the problem first.

Problem 1. Input is a sequence of updates S_t followed by a sequence S'_t .

Output: Let \mathbf{a} be the state of sequence S_t , and let \mathbf{a}' be the state of S'_t . Output $\mathbf{a} \cdot \mathbf{a}' \pm \alpha = \sum_{i \in \mathcal{U}} a_i a'_i \pm \alpha$, where α is an approximation factor.

Theorem 9. *Let \mathcal{A} be a streaming algorithm that on input streams S_t, S'_t outputs $\mathbf{a} \cdot \mathbf{a}' \pm o(\sqrt{m})$ with probability at least $1 - O(m^{-2})$. Then \mathcal{A} is not ϵ -pan private for any constant ϵ .*

Proof. Fix a stream S_t s.t. $\forall i \in \mathcal{U} : a_i \in \{0, 1\}$. Let the internal state of the algorithm \mathcal{A} after processing S_t be X . By the definition of pan privacy, I is ϵ -differentially private with respect to S_t . Fix some constants δ and η . We will show that for all large enough m , any algorithm \mathcal{Q} that takes as input X and a stream S'_t and outputs $\mathbf{a} \cdot \mathbf{a}' \pm o(\sqrt{m})$ with probability at least $1 - O(m^{-2})$ can be used to recover a_i exactly for all but an η fraction of $i \in \mathcal{U}$ with probability $1 - \delta$. Therefore, the existence of such an algorithm \mathcal{Q} implies that X cannot be ϵ -differentially private for any fixed ϵ . Indeed, assume for the sake of contradiction that an algorithm with the given properties exists and X is ϵ -differentially private. Since \mathcal{Q} depends only on X and not on S_t , the output of \mathcal{Q} is also ϵ -differentially private. This is a contradiction, since the output of \mathcal{Q} can be used to guess a bit of the binary vector \mathbf{a} accurately with probability at least $(1 - 2\delta - \eta)$, where δ and η can be chosen arbitrarily small.

To finish the proof we show that an algorithm \mathcal{Q} with the specified properties can be used to recover all but an η fraction of \mathbf{a} with probability $1 - \delta$. To see this, observe that \mathcal{Q} can be used to answer queries $\mathbf{a} \cdot \mathbf{q}$ for any arbitrary \mathbf{q} to within $o(\sqrt{m})$ additive error. In particular, to answer queries $\mathbf{a} \cdot \mathbf{q}_1, \dots, \mathbf{a} \cdot \mathbf{q}_r$, run $\mathcal{Q}(X, S_t^{(1)}), \dots, \mathcal{Q}(X, S_t^{(r)})$ in parallel, where $S_t^{(i)}$ is a stream with state \mathbf{q}_i . If $r = o(n^2)$, then, by the union bound, with probability $1 - \delta$ for any constant δ , $\mathcal{Q}(X, S_t^{(i)}) = \mathbf{a} \cdot \mathbf{q}_i \pm o(\sqrt{m})$. By Theorem 6, there exists an algorithm that, given the output of $\mathcal{Q}(X, S_t^{(1)}), \dots, \mathcal{Q}(X, S_t^{(r)})$, outputs $\tilde{\mathbf{a}}$ s.t. except with negligible probability $\tilde{\mathbf{a}}$ agrees with \mathbf{a} on all but η fraction of the coordinates. \square

Notice that the lower bound relies on the fact that the updates for S_t arrive before any of the updates of S'_t . This restriction can be relaxed. In general, we get a lower bound of $\Omega(\sqrt{m_0})$ for the additive error, where m_0 is the largest number of items in S that are updated before any of the corresponding items in S' . The lower bound is interesting whenever the updates to the two sequences of updates are not “synchronized”, i.e. $(i, d) \in S_t$ (i, d') $\in S'_t$ for the same $i \in \mathcal{U}$ are allowed to arrive at different time steps.

Recall that the distinct count for S_t is $D^{(t)}$. We have the following corollary.

Corollary 1. *Let \mathcal{A} be an online algorithm that on input S_t outputs $D^{(t)} \pm o(\sqrt{m})$ with probability at least $1 - O(m^{-2})$. Then \mathcal{A} is not ϵ -pan private for any constant ϵ .*

Proof. Notice that the proof of Theorem 9 goes through if we restrict the instances to be binary, i.e. if we require that $\forall i \in [m] : \mathbf{a}, \mathbf{a}' \in \{0, 1\}$. The corollary follows by a reduction from this restricted dot-product problem to the distinct elements problem. Given binary streams S'_t, S''_t , let $S_t = (S'_t, S''_t)$ be their concatenation. By a simple application of inclusion-exclusion, $D^{(t)} = D^{(t)}(S') + D^{(t)}(S'') - \mathbf{a} \cdot \mathbf{a}'$. Therefore, an ϵ -pan private algorithm for $D^{(t)}$ that achieves additive approximation α with probability $1 - \delta$ implies a 3ϵ -pan private algorithm for dot product on binary instances that achieves additive approximation 3α with probability $1 - 3\delta$. \square

The next two theorems follow by arguments identical to the one used to prove Theorem 9, but using, respectively, Theorem 7 and Theorem 8 in place of Theorem 6.

Theorem 10. *Let \mathcal{A} be an online algorithm that on inputs S_t, S'_t outputs $\mathbf{a} \cdot \mathbf{a}' \pm o(\sqrt{m})$ with probability at least $1 - \delta$. If $\delta < \rho^*/2(1 + \eta)$ for any η , then \mathcal{A} is not ϵ -pan private for any constant ϵ .*

Proof. The proof is analogous to the proof of Theorem 9. Note first that the $\{-1, 1\}$ queries of Theorem 7 can be simulated as the difference of two $\{0, 1\}$ queries, which gives $o(\sqrt{m})$ additive error with probability at most $1 - 2\delta$. In order to apply Theorem 7, we need to guarantee that at most $\rho < \rho^*$ fraction of the queries answered by \mathcal{Q} have error $\Omega(\sqrt{m})$. Call such queries *inaccurate*. In expectation there are at most 2δ inaccurate queries. Since the statement of Theorem 7 holds when the queries are independent, an application of a Chernoff bound with a large enough number of queries shows that except with negligible probability there are at most ρ^* inaccurate queries. After applying Theorem 7 the proof can be finished analogously to the proof of Theorem 9. \square

Corollary 2. *Let \mathcal{A} be an online algorithm that on input S outputs $D^{(t)}(S) \pm o(\sqrt{m})$ with probability at least $1 - \delta$. If $\delta < \rho^*/6(1 + \eta)$, then \mathcal{A} is not ϵ -pan private for any constant ϵ .*

This corollary implies the optimality of the full-space distinct counts estimation algorithm presented in Section 4 *when only additive approximations are allowed*.

Using similar arguments, we can show the following (proof omitted).

Theorem 11. *Let \mathcal{A} be a streaming algorithm that on input a stream S_t and any constant α outputs $(1 \pm \alpha)D^{(t)} \pm o(\log^c m)$ with probability at least $1 - n^{-\Omega(\log^c m)}$. Then \mathcal{A} is not ϵ -pan private for any constant ϵ .*

The theorem establishes that when an arbitrarily small multiplicative approximation factor is allowed, an additive polylogarithmic error is unavoidable for the problem of estimating distinct counts. Thus, up to the exact order of the polylogarithmic additive factor, our sketching algorithm for distinct count estimation is optimal.

4 Heavy Hitters

We provide further evidence for the usefulness of sketching for pan-private algorithms by presenting an improved algorithm for the Heavy Hitters problem. As a tool we use a variant of the cropped mean estimator from [4], but we combine it with a sketching approach in the style of CM sketches [1], instead of the sampling approach used in [4]. This will allow us to significantly reduce the approximation error by reducing the universe size while approximately preserving the number of heavy items. Once again, we observe that polylogarithmic space complexity is a by-product of the improved approximation ratio.

4.1 Full Space Cropped Sum

We begin with an analysis of the cropped sum estimator in full space. For completeness we describe the estimator. We will approximate $T_k^{(t)}(\tau)$ for a universe \mathcal{U} and a sequence of updates S_t .

Let \mathcal{D}_0 be the uniform distribution over $\{0, 1\}$ and \mathcal{D}_1 be the distribution that assigns probability $1/2 + \epsilon/4$ to 1 and the remaining probability to 0. We compute an estimate $\tilde{T}_k(\tau)$ of $T_k(\tau)$ as follows:

- For each $j \in \mathcal{U}$, initialize a counter $c_j \in_R \{0, \dots, \tau - 1\}$, a bit $b_j \sim \mathcal{D}_0$
- When item j arrives on the stream, increment the counter $c_j \pmod{\tau}$. If $c_j = 0$ pick b_j from \mathcal{D}_1 .
- At query time, compute $o := |\{j : b_j = 1\}|$, and output $\tilde{T}(\tau) = (o - |\mathcal{U}|/2) \frac{4\tau}{\epsilon}$.

Note that this algorithm is simply an instantiation of the cropped mean estimator from [4] in full space. Keeping counters for each element allows us to guarantee smaller additive error in terms of m .

Lemma 5. *The estimator $\tilde{T}_k(\tau)$ is ϵ -differentially private. Moreover, with probability $1 - 2e^{-2\alpha}$,*

$$|T_k(\tau) - \tilde{T}_k(\tau)| \leq \frac{4\alpha t \sqrt{|\mathcal{U}|}}{\epsilon}.$$

Proof. The privacy analysis is identical to the privacy analysis of the cropped mean estimator in [4]. By the analysis in [4], $\mathbb{E}[o] = |\mathcal{U}|/2 + \epsilon T_k(\tau)/4\tau$, and, therefore, $\mathbb{E}[\tilde{T}_k(\tau)] = T_k(\tau)$. By Hoeffding's bound, $\Pr[|o - \mathbb{E}[o]| \geq \alpha \sqrt{|\mathcal{U}|}] \leq 2e^{-2\alpha}$. The lemma follows. \square

Note that setting the cropping parameter t to 1 gives an estimate of the distinct count D in full space with $O(\sqrt{m})$ additive error.

4.2 HH Algorithm

The limiting factor in the cropped sum estimator is m . Even though we allow full space to the algorithm and it achieves pan-privacy, the approximation guarantees involve an additive factor in m which is large. The key step in our algorithm is to project the input S onto S' over a much smaller universe, so that S' has approximately the same k -heavy hitters count. In fact, we are able to reduce the universe size to a constant that depends only on k and the desired approximation guarantee. The reduced universe size directly implies

a more accurate cropped sum estimate and, hence, a more accurate estimate of the number of k -heavy hitters. Next we present our algorithm.

Assume the value $F_1 = F_1^{(t_0)}$, where t_0 is the time step when the algorithm will be queried, is known ahead of time. Assume also we have oracle access to a random function $f : [m] \rightarrow [h]$ (these assumptions will be removed in Section 5). Given a sequence of updates S , let $f(S)$ be the sequence $(f(i_1), d_1), \dots, (f(i_t), d_t)$, and let $T_k(\tau|f)$ and $\tilde{T}_k(\tau|f)$ be, respectively, $T_k(\tau)$ and $\tilde{T}_k(\tau)$ computed on the stream $f(S)$. Note that $f(S)$ is a stream over the universe $[h]$ and can easily be simulated online given the oracle for f .

- Choose a random function $f : \mathcal{U} \rightarrow [h]$. Compute $x_1 = \tilde{T}_k(F_1/k|f)$ and $x_2 = \tilde{T}_k(F_1/c_k|f)$. Output

$$\tilde{HH}(k) := (x_1 - x_2) \left(\frac{F_1}{k} - \frac{F_1}{c_k} \right)^{-1}$$

The above algorithm will be accurate provided that the function f approximately preserves the number of heavy hitters. In the next section we show that a random f satisfies this condition with high probability.

4.3 Reducing the Universe Size

Remember that we denote $a_i = \sum_{j:i_j=i} d_j$.

Lemma 6. *Let $f : \mathcal{U} \rightarrow [h]$ be a random function. Also, let $\tilde{k} = |\{j : \exists i \in h^{-1}(j) \text{ s.t. } a_i \geq t/k\}|$. With probability $1 - \delta$,*

$$\frac{\tilde{k}}{HH(k)} \geq 1 - \frac{k}{\delta h}.$$

Proof. Let the indicator random variable I_j be equal to 1 iff $\forall i \in h^{-1}(j) : a_i < t/k$. The expected value of I_j for any j is as follows:

$$\mathbb{E}[I_j] = \left(1 - \frac{1}{h}\right)^{HH(k)} \leq \exp(-HH(k)/h).$$

Denote, for convenience, $r := h/HH(k)$. We can write \tilde{k} in terms of I_j :

$$\begin{aligned} \mathbb{E}[\tilde{k}] &= \sum_{j \in [h]} (1 - I_j) \geq h(1 - e^{-1/r}) \\ \mathbb{E}\left[\frac{\tilde{k}}{HH(k)}\right] &\geq r(1 - e^{-1/r}). \end{aligned}$$

Using the inequality $e^x \geq 1 + x + x^2$ (valid for $x \in [-1, 1]$), we simplify to

$$\mathbb{E}\left[\frac{\tilde{k}}{HH(k)}\right] \geq r\left(1 - 1 + \frac{1}{r} - \frac{1}{r^2}\right) = 1 - \frac{1}{r}$$

We can apply Markov's inequality to the random variable $(HH(k) - \tilde{k})/HH(k) > 0$. Therefore, with probability $1 - \delta$,

$$\frac{\tilde{k}}{HH(k)} \geq 1 - \frac{1}{\delta r} \geq 1 - \frac{k}{\delta h}.$$

□

In the next lemma we show that we can project the universe onto a significantly smaller universe without creating “new” heavy hitters.

Lemma 7. *Let $A \subseteq \mathcal{U}$ be set of items s.t. $\forall i \in A : a_i \leq F_1\delta/2k^2$. Also, let $f : \mathcal{U} \rightarrow [h]$ be a pairwise-independent hash function. There exists an $h_0 = \Theta(k)$, s.t. for any $h \geq h_0$ with probability at least $1 - \delta$*

$$\forall j \in [h] : \sum_{i \in A \cap f^{-1}(j)} a_i \leq F_1/k.$$

Proof. Let $N_j = \sum_{i \in A \cap f^{-1}(j)} a_i$, i.e. N_j is the total frequency of the items mapped to j by f . It's easy to see that $\mathbb{E}[N_j] = \frac{1}{h} \sum_{i \in A} a_i \leq F_1/h$. Let's analyze the variance. Let X_{ij} be the indicator variable for the event $\{i \in B_j\}$. By pairwise independence, $\text{Var}(N_j) = \sum_{i \in A} \text{Var}(X_{ij}a_i)$.

$$\begin{aligned} \text{Var}(X_{ij}a_i) &= \mathbb{E}[(X_{ij}a_i)^2] - \mathbb{E}[X_{ij}a_i]^2 \\ &= a_i^2 \left(\frac{1}{h} - \frac{1}{h^2} \right). \end{aligned}$$

Therefore, $\text{Var}(N_j) = \left(\frac{1}{h} - \frac{1}{h^2} \right) \sum_{i \in A} a_i^2$. We will denote $\sum_{i \in A} a_i^2$ as $F_2(A)$.

Fact 1. If $\sum_{i \in A} a_i \leq F_1$ and $\forall i \in A : a_i \leq pF_1$ for some $p \in [0, 1]$, $F_2(A) = \sum_{i \in A} a_i^2 \leq p(F_1)^2$.

Proof. Let \mathbf{a} be a vector that maximizes $F_2(A)$. We may assume without loss of generality that $\sum_{i \in A} a_i = F_1$. Then either 0 or at least two coordinates in \mathbf{a} can be in the open interval $(0, pF_1)$. We claim that there exists a maximum \mathbf{a} s.t. all coordinates are equal to either 0 or pF_1 . Assume, for contradiction, that there exist i and i' in A s.t. $0 < a_i < pF_1$ and $0 < a_{i'} < pF_1$. Let $a_i \geq a_{i'}$. Then changing a_i to $a_i + 1$ and $a_{i'}$ to $a_{i'} - 1$ strictly increases $F_2(A)$ which is a contradiction. \square

By Fact 1, $F_2(A) \leq (F_1)^2 \delta / 2k^2$. Set $h \geq h_0 = (\sqrt{2} + 2)k$. By the one-sided Chebyshev inequality,

$$\begin{aligned} \Pr[N_j \geq \frac{F_1}{k}] &\leq \frac{1}{1 + \left(\frac{F_1}{k} - \frac{F_1}{h} \right)^2 / \left(\frac{1}{h} - \frac{1}{h^2} \right) F_2(A)} \\ &< \frac{F_2(A)}{h \left(\frac{F_1}{k} - \frac{F_1}{h} \right)^2} \leq \frac{\delta}{2hk^2 \left(\frac{1}{k} - \frac{1}{h} \right)^2} = \frac{\delta}{2h \left(1 - \frac{1}{\sqrt{2}+2} \right)^2} = \frac{\delta}{h}. \end{aligned}$$

The lemma follows by a union bound. \square

We are now ready to analyze $\tilde{\text{HH}}(k)$. The following theorem shows that $\tilde{\text{HH}}(k)$ is in the range $[(1 - \beta) \text{HH}(k) - O(\sqrt{k}), \text{HH}(O(k^2)) + O(\sqrt{k})]$ with constant probability.

Theorem 12. $\tilde{\text{HH}}(k)$ can be computed while satisfying 2ϵ -pan privacy. Moreover, if $h \geq \max\{k/\beta\delta, (\sqrt{2} + 2)ck\}$, then with probability $1 - 2\delta - 4\exp(-\alpha)$

$$(1 - \beta) \text{HH}(k) - \frac{4(c+1)\alpha\sqrt{h}}{(c-1)\epsilon} \leq \tilde{\text{HH}}(k) \leq \text{HH}(2c^2k^2/\delta) + \frac{4(c+1)\alpha\sqrt{h}}{(c-1)\epsilon}.$$

Proof. The privacy guarantee follows by the ϵ -pan privacy of the cropped sum estimators and the composition theorem of Dwork et al. [6]. Next we analyze utility.

Computing cropped F_1 at two levels of the cropping parameter gives us an approximation of the number of heavy hitters:

$$\begin{aligned} T_1(F_1/k) - T_1(F_1/ck) &= \sum_{j: N_j \geq F_1/ck} \min(N_j, F_1/k) - F_1/ck \\ &= \sum_{j: N_j \geq F_1/k} (F_1/k - F_1/ck) + \sum_{j: F_1/ck \geq N_j \geq F_1/k} (N_j - F_1/ck) \end{aligned}$$

It immediately follows that $|\{j : N_j \geq F_1/k\}| < \mathbb{E}[\tilde{\text{HH}}(k)] \leq |\{j : N_j \geq F_1/ck\}|$. By Lemma 6, $|\{j : N_j \geq F_1/k\}| \geq (1 - \beta) \text{HH}(k)$ except with probability δ . We can apply Lemma 7 with $A = \{i : a_i \leq F_1\delta/2c^2k^2\}$. By the lemma, for every $j \in [h]$ we have $N_j \geq F_1/ck \Rightarrow \exists i \in f^{-1}(j)$ s.t. $a_i \geq F_1\delta/(2c^2k^2)$, except with probability δ . Therefore, $|\{j : N_j \geq F_1/ck\}| \leq \text{HH}(c^2k^2/\delta)$. We have thus shown that

$$(1 - \beta) \text{HH}(k) \leq (T_1(F_1/k|f) - T_1(F_1/ck|f)) \left(\frac{F_1}{k} - \frac{F_1}{ck} \right)^{-1} \leq \text{HH}(c^2k^2/\delta).$$

With probability $1 - 4e^{-2\alpha}$, Lemma 5 gives us the following guarantees:

$$\begin{aligned}\Pr[|\tilde{T}_1(F_1/k|f) - T_1(F_1/k|f)| > \frac{4\alpha\sqrt{ht}}{k\epsilon}] &\leq 2\exp(-\alpha) \\ \Pr[|\tilde{T}_1(F_1/ck|f) - T_1(F_1/ck|f)| > \frac{4\alpha\sqrt{ht}}{ck\epsilon}] &\leq 2\exp(-\alpha)\end{aligned}$$

With probability $1 - 4e^{-2\alpha}$,

$$|(x_1 - x_2) - (T_1(F_1/k|f) - T_1(F_1/ck|f))| \leq \frac{4\alpha\sqrt{ht}}{\epsilon k} \left(1 + \frac{1}{c}\right).$$

A straightforward computation and a union bound will complete the proof. \square

In previous work Dwork et al. [4] present an algorithm that outputs an estimate for $\text{HH}(k)$ in $[\text{HH}(k(1 + \rho)) - \alpha * m, \text{HH}(k/(1 + \rho)) + \alpha * m]$ with probability $1 - \delta$. Extending their algorithm to full space can improve the additive error to $O(\sqrt{m})$ with constant probability. For $k = O(1)$, which is the usual range for this parameter, our algorithm outperforms Dwork et al.'s.

5 Extensions

In the following we extend ideas used in the Heavy Hitters upper bound to other problems. We consider the cropped second moment and inner product problems which have not been addressed in the context of pan-privacy before. The performance of our inner product algorithm matches the lower bound presented in Section 3.4. We show how to relax some of the assumptions made in Section 4.

5.1 Inner Products and T_2

A simple extension of the cropped sum estimator from Section 4 allows us to estimate the cropped dot product of two sets of updates, as well as the cropped second moment of an input.

Let, as before, \mathcal{D}_0 be the uniform distribution over $\{0, 1\}$ and \mathcal{D}_1 be the distribution that assigns probability $1/2 + \epsilon/4$ to 1 and the remaining probability to 0. We compute an estimate $\widetilde{(\mathbf{a} \cdot \mathbf{a}')(\tau)}$ of $(\mathbf{a} \cdot \mathbf{a}')(\tau)$ as follows:

- For each $i \in [m]$, initialize a independently initialized counters $c_i, c'_i \in_R \{0, \dots, \sqrt{\tau} - 1\}$, and bits $b_i, b'_i \sim \mathcal{D}_0$
- When item i arrives as an update in S , increment the counter $c_i \pmod{\sqrt{\tau}}$. If $c_i = 0$ pick b_i from \mathcal{D}_1 . Process the updates in S' analogously.
- At query time,
 - compute $o := |\{i : b_i = b'_i = 1\}|$;
 - output:

$$\widetilde{(\mathbf{a} \cdot \mathbf{a}')(\tau)} = (o - \tilde{T}_1(\sqrt{\tau})/2 - \tilde{T}'_1(\sqrt{\tau})/2 - m/4) \frac{16\tau}{\epsilon^2}$$

Lemma 8. *The estimator $\widetilde{(\mathbf{a} \cdot \mathbf{a}')(\tau)}$ is 2ϵ -differentially private. Moreover, with probability $1 - 6e^{-2\alpha}$,*

$$|(\mathbf{a} \cdot \mathbf{a}')(\tau) - \widetilde{(\mathbf{a} \cdot \mathbf{a}')(\tau)}| \leq \frac{16\alpha\tau\sqrt{m}}{\epsilon^2} \left(1 + \frac{\epsilon}{4\sqrt{\tau}}\right).$$

Proof. The proof of privacy follows from the analysis of the cropped means estimator [4]. The utility analysis is also a simple extensions as follows.

By the analysis of [4], for every $i \in [m]$, $\Pr[b_i = 1] = 1/2 + \epsilon \min(a_i, \sqrt{\tau})/4\sqrt{\tau}$, and similarly $\Pr[b'_i = 1] = 1/2 + \epsilon \min(a'_i, \sqrt{\tau})/4\sqrt{\tau}$. Since for every i , b_i and b'_i are independent, we have

$$\begin{aligned} \Pr[b_i = b'_i = 1] &= \left(\frac{1}{2} + \frac{\epsilon \min(a_i, \sqrt{\tau})}{4\sqrt{\tau}} \right) \left(\frac{1}{2} + \frac{\epsilon \min(a'_i, \sqrt{\tau})}{4\sqrt{\tau}} \right) \\ &= \frac{1}{4} + \frac{\epsilon \min(a_i, \sqrt{\tau})}{8\sqrt{\tau}} + \frac{\epsilon \min(a'_i, \sqrt{\tau})}{8\sqrt{\tau}} + \frac{\epsilon^2 \min(a_i a'_i, \tau)}{16\tau}. \end{aligned}$$

Therefore, $\mathbb{E}[\widetilde{(\mathbf{a} \cdot \mathbf{a}')(\tau)}] = (\mathbf{a} \cdot \mathbf{a}')(\tau)$, and the theorem follows by a Hoeffding bound and the guarantees for \tilde{T}_1 . \square

Notice that Lemma 8 is valid regardless of whether S_t and S'_t are interleaved in an arbitrary manner. Also, we can take $S_t = S'_t$, and the algorithm gives an estimate for T_2 , i.e. $\tilde{T}_2(\tau) = \widetilde{(\mathbf{a} \cdot \mathbf{a})(\tau)}$.

5.2 Random Oracle and $F_1^{(t_0)}$

Two assumptions that we make in Section 4 are that we have oracle access to a random function f and that the value $F_1^{(t_0)}$ at the time step t_0 when the algorithm is queried is known before the sequence of updates is processed. Here we show how these assumptions can be relaxed.

Notice that, assuming a bound $a_i \leq U$, our heavy hitters algorithm uses constant space. Therefore Nisan's pseudorandom generator [10] can be used to remove the first assumption. To address the second assumption, we can assume an upper bound U_0 on $F_1 = F_1^{(t_0)}$. Then we can run $\log U_0 + 1$ instances of our heavy hitters algorithm in parallel with F'_1 (the projected value of $F_1^{(t_0)}$ set to $1, 2, 4, \dots, U_0$, respectively. At query time we use the output of the algorithm instance with F'_1 set to $2^{\lceil \log F_1 \rceil}$. This procedure gives us a $2(\log U_0 + 1)\epsilon$ -pan private algorithm that outputs an estimate $\tilde{\text{HH}}(k) \in [(1 - \beta) \text{HH}(k) - O(\sqrt{k}), \text{HH}(O(k^2)) + O(\sqrt{k})]$.

6 Concluding Remarks

Inspired by [4], we study pan-private algorithms that guarantee differential privacy of data analyses even when the internal memory of the algorithm may be compromised by an unannounced intrusion of an attacker. [4] used techniques from random response [13] on top of sampling to get pan-private streaming algorithms for some of the basic statistical estimates on the input.

We addressed fundamental questions about the memory, its size and its role in pan-privacy. We showed that distinct count can not be estimated accurately to additive error even given unbounded space; this is based on approach for showing lower bounds via noisy decoding. We also showed a streaming algorithm that is pan-private and matches this accuracy. We also show worst case $O(k)$ approximate streaming pan-private algorithm for estimating heavy-hitter counts. Both of these upper bounds come from using sketches. Also, it is interesting that while we do not require pan-private algorithms to use small memory, the best known algorithms so far are streaming, that is, they use sublinear memory.

We find the notion of pan-privacy to be intriguing, and believe more needs to be understood in this intersection of differential privacy and streaming. For example, in streaming, many problems can be solved in presence negative and positive d_j 's. While our distance count estimation in this paper works in this case and is pan-private, we leave it open to address the difficulty of obtaining pan-private algorithms for other problems in such cases. Also, the basic model of pan-privacy here can be extended to the case when there are multiple intrusions or even continual intrusions [5]. Under those models, what statistical estimates can be computed accurately and privately?

We conclude this paper with the observation that our insights so far give pan-private approximations for related problems such as T_2 (cropped F_2) and inner products. We leave it open to extend these results to other problems such as entropy estimation.

References

- [1] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.
- [2] Graham Cormode, Mayur Datar, Piotr Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Transactions on Knowledge and Data Eng.*, 15(3):529–540, 2003.
- [3] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS '03: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, New York, NY, USA, 2003. ACM.
- [4] C. Dwork, M. Naor, T. Pitassi, G. Rothblum, and S. Yekhanin. Pan-Private Streaming Algorithms. In *ICS'10: Innovations In Computer Science Conference*, 2010.
- [5] Cynthia Dwork. Differential privacy in new settings. In *SODA'10: ACM-SIAM Symposium On Discrete Algorithms*. ACM-SIAM, 2010.
- [6] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC'06: In Proceedings of the 3rd Theory of Cryptography Conference*, 2006.
- [7] Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of lp decoding. In *STOC'07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 85–94, New York, NY, USA, 2007. ACM.
- [8] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. In *Journal of ACM*, pages 307–323. ACM, 2006.
- [9] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in TCS*, 2(1):1–113, 2005.
- [10] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.
- [11] J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, Boston, 2010. In progress, Chapter 1 online at academic2.american.edu/~jpnolan.
- [12] Kunal Talwar, Anupam Gupta, Katrina Ligett, Frank McSherry, and Aaron Roth. Differentially private combinatorial optimization. In *SODA'10: ACM-SIAM Symposium On Discrete Algorithms*. ACM-SIAM, 2010.
- [13] S. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of American Statistical Association*, (60):63–69, 1965.